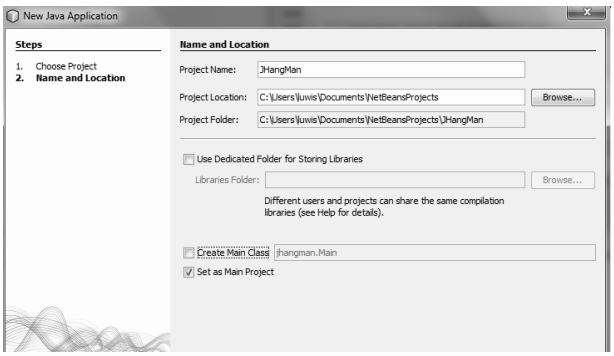
2

APLIKASI GAME HANGMAN

Pada bab ini kita akan membuat aplikasi game Hang Man. Game ini adalah game tebak kata. Pemain akan menebak kata yang tersembunyi huruf demi huruf. Pemain diberi kesempatan menebak hingga 6 kali kesalahan. Jika kesempatan menebak habis, pemain akan digantung. Jika berhasil menebak kata yang dimaksud, pemain selamat.

MEMBUAT PROJECT

1. Buat project baru, beri nama **JHangMan**.



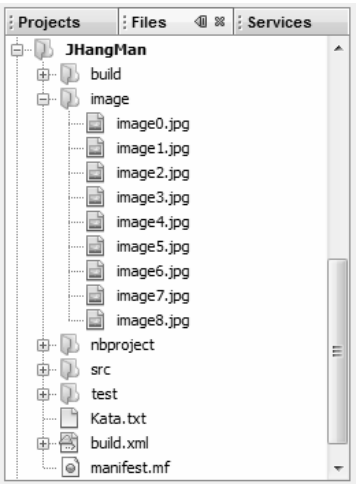
Gambar 2.1

Membuat project baru.

2. Copy file **Kata.txt** (**CD:\Source\bab1\Kata.txt**) pada direktori utama project. File Kata.txt ini berisi kata-kata untuk ditebak.
3. Buat folder baru. Klik kanan pada project **JHangMan > New > Folder**. Beri nama **image** dan tempatkan pada direktori utama project.
4. Copy semua file image (**CD:\Source\bab1\image**) pada folder image yang telah dibuat.
5. Susunan files project akan menjadi seperti ini.

Gambar 2.2

Susunan file project.



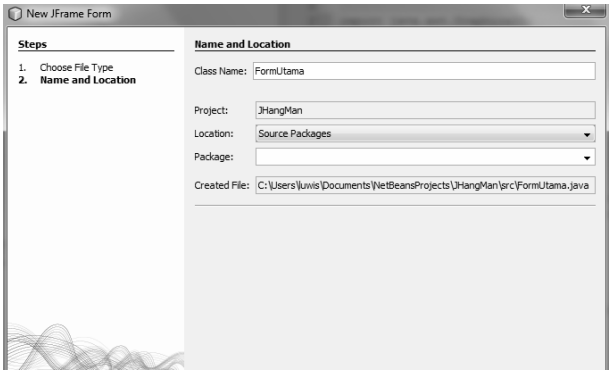
6. Sertakan folder icon tadi dalam project. Klik kanan project **JHangMan** > pilih Categories: **Sources** > **Sources package folders** > **Add folder** > pilih forder image.

MENDESAIN GUI

1. Buat sebuah JFrame. Klik kanan pada project **JHangMan** > **New** > **JFrame Form**. Beri nama **FormUtama**.

Gambar 2.3

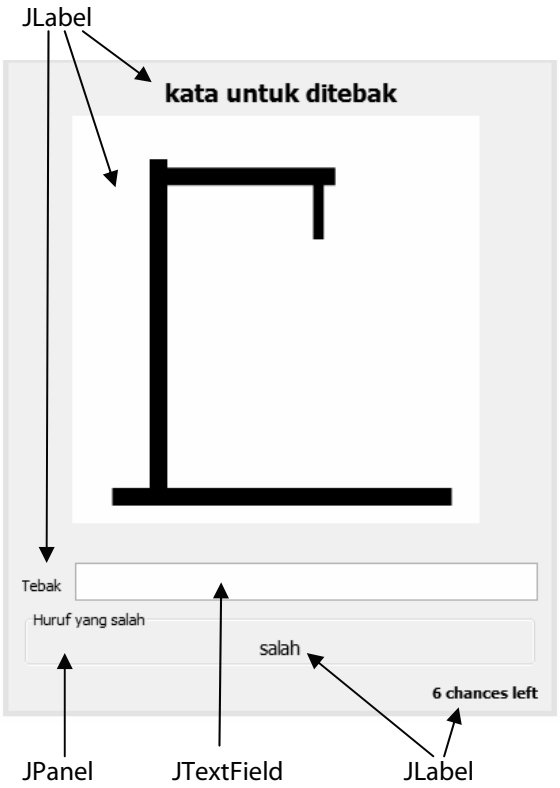
Membuat FormUtama.



2. Tambahkan **5 JLabel**, **1 JTextField**, dan **1 JPanel** pada FormUtama. Atur tampilan seperti ini.

Gambar 2.4

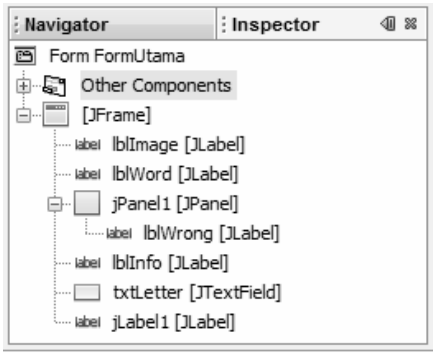
Design GUI FormUtama.



3. Susunan komponen FormUtama pada window inspector menjadi seperti ini.

Gambar 2.5

Susunan komponen FormUtama.



4. Atur propertiesnya.

FormUtama

Enabled	False
Title	Game HangMan

JPanel1

Border		Titled Border
	Border	Huruf yang salah

JTextField1

Variable Name	txtLetter
Text	-
Font	Tahoma 18 Plain

JLabel1

Text	Tebak

JLabel2

Variable Name	lblWord
Text	Kata Untuk ditebak
Horizontal Alignment	CENTER
Font	Tahoma 18 Bold

JLabel3

Variable Name	lblImage
Text	-
Icon	Image0.jpg

JLabel4

Variable Name	lblWrong
Text	salah
Font	Tahoma 14 Plain
Horizontal Alignment	CENTER

JLabel5

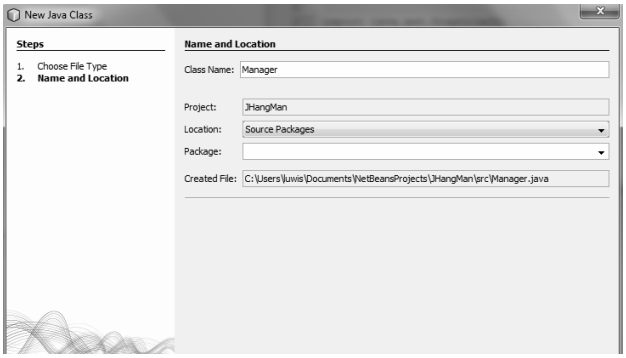
Variable Name	lblInfo
Text	6 chances left
Font	Tahoma 11 Bold

MEMBERI ACTION

1. Buat class baru. Klik kanan pada project **JHangMan > New > Java Class**. Beri nama **Manager**. Class ini akan mengurusi masalah mengenai pengambilan data dari file Kata.txt serta mengacak kata untuk ditebak.

Gambar 2.6

Membuat class Manager.



2. Ketikkan code berikut pada Manager.java.

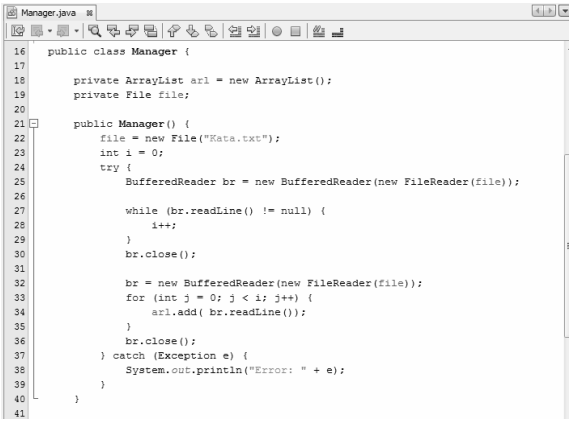
```
public class Manager {
private ArrayList arl = new ArrayList();
private File file;

public Manager() {
file = new File("Kata.txt");
int i = 0;
try {
BufferedReader br = new BufferedReader(new
FileReader(file));
while (br.readLine() != null) {
i++;
}
br.close();
br = new BufferedReader(
                    new FileReader(file));
for (int j = 0; j < i; j++) {
arl.add( br.readLine());
}
br.close();
} catch (Exception e) {
System.out.println("Error: " + e);
}
}

public String getRandomWord()
                        throws Exception {
int random =
      (int) (Math.random() * arl.size());
return (String) arl.get(random);
}
}
```

Gambar 2.7

Code program class Manager.java.



Program membaca semua kata dari file Kata.txt. Seluruh kata tersebut ditampung dalam sebuah arraylist. Method getRandomWord() akan mengambil salah satu kata yang ada dalam arraylist tersebut secara acak.

3. Kembali lagi ke FormUtama. Klik tab SourceTambah variable berikut.

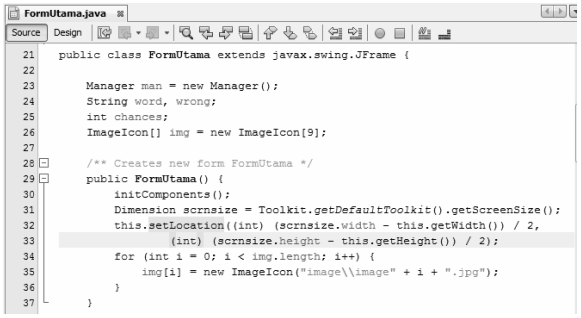
```
Manager man = new Manager();
String word, wrong;
int chances;
ImageIcon[] img = new ImageIcon[9];
```

4. Atur supaya posisi aplikasi saat tampil berada di tengah-tengah monitor saat pertama kali aplikasi dijalankan. Ketik code berikut pada constructor FormUtama.

```
public FormUtama() {
initComponents();
Dimension scrnsize = Toolkit.
         getDefaultToolkit().getScreenSize();
this.setLocation((int)
      (scrnsize.width - this.getWidth()) / 2,
      (int)
   (scrnsize.height - this.getHeight()) / 2);
for (int i = 0; i < img.length; i++) {
img[i] = new ImageIcon(
                "image\\image" + i + ".jpg");
}
}
```

Gambar 2.8

Variable dan constructor FormUtama.



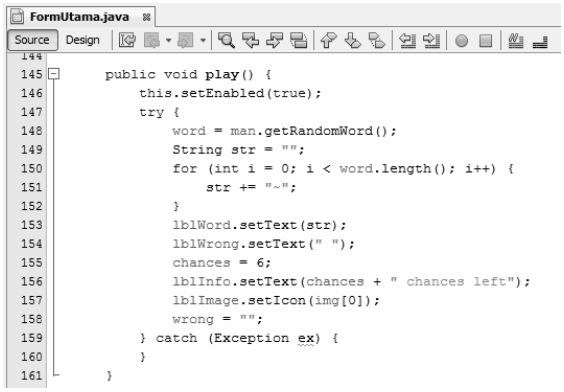
5. Buat method play(). Method ini akan mengeset semua nilai variable awal ketika permainan dimulai dan mengambil kata secara acak untuk ditebak.

```
public void play() {
this.setEnabled(true);

try {
word = man.getRandomWord();
String str = "";
for (int i = 0; i < word.length(); i++) {
str += "~";
}
lblWord.setText(str);
lblWrong.setText(" ");
chances = 6;
lblInfo.setText(chances + " chances left");
lblImage.setIcon(img[0]);
wrong = "";
} catch (Exception ex) {
}
}
```

Gambar 2.9

Code method play().



6. Buat method guessWord(). Method ini akan melakukan semua hal yang berkaitan dengan proses penebakan kata.

```
public void guessWord() {
String letter = txtLetter.getText();
char ltr = letter.toLowerCase().charAt(0);

String guess = "";
for (int i = 0; i < word.length(); i++) {
if (ltr == word.charAt(i)) {
guess += ltr;
} else {
if (lblWord.getText().charAt(i) == '~') {
guess += "~";
} else {
guess += lblWord.getText().charAt(i);
}
}
}
if (lblWord.getText().equals(guess)) {
int x = 7 - chances;
wrong += " " + ltr;
if (chances > 0) {
chances--;
lblImage.setIcon(img[x]);
lblInfo.setText(chances + " chances left");
lblWrong.setText(wrong);
txtLetter.setText("");
txtLetter.setFocusable(true);
} else {
this.setEnabled(false);
lblImage.setIcon(img[7]);
int lanjut =
      JOptionPane.showOptionDialog(this,
      "Kata yang benar adalah " + word +
      ". Ulang permainan?",
      "Anda digantung!!!",
      JOptionPane.YES_NO_OPTION,
      JOptionPane.QUESTION_MESSAGE,
      null, null, null);
if (lanjut == JOptionPane.YES_OPTION) {
play();
txtLetter.setText("");
txtLetter.setFocusable(true);
} else {
System.exit(0);
}
}
} else {
if (word.equals(guess)) {
lblWord.setText(guess);
lblImage.setIcon(img[8]);
this.setEnabled(false);
int lanjut =
      JOptionPane.showOptionDialog(this,
      "Lanjut permainan?", "Anda selamat!!!",
      JOptionPane.YES_NO_OPTION,
      JOptionPane.QUESTION_MESSAGE,
      null, null, null);
if (lanjut == JOptionPane.YES_OPTION) {
play();
txtLetter.setText("");
txtLetter.setFocusable(true);
} else {
System.exit(0);
}
} else {
```

```
lblWord.setText(guess);
txtLetter.setText("");
txtLetter.setFocusable(true);
}
}
}
```

User menginput sebuah huruf untuk menebak kata yang dimaksud. Jika user menginput lebih dari 1 huruf maka hanya huruf pertama yang digunakan. Kemudian program akan memeriksa huruf yang diinput apakah ada di dalam kata yang maksud. Apakah huruf yang diinput tadi menambah jumlah kata yang berhasil ditebak atau tidak. Jika tebakan salah, kesempatan berkurang 1 dan jika kesempatan habis, permainan selesai dan program akan menampilkan kata yang dimaksud. Jika tebakan benar dan semua kata sudah tertebak maka permainan selesai.

7. Implementasikan method guessWord(), beri action pada txtLetter. Klik kanan **txtLetter > Event > action > actionPerformed**.

Gambar 2.10

Memberi action pada txtLetter.

8. Ketik code berikut.

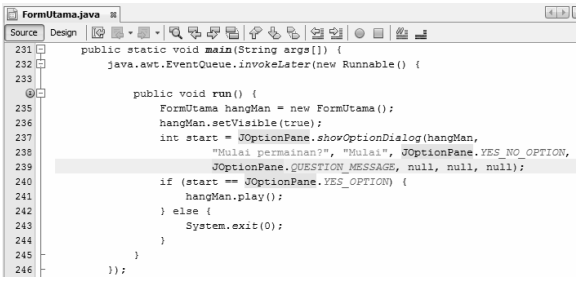
```
private void txtLetterActionPerformed(
        java.awt.event.ActionEvent evt) {
guessWord();
}
```

9. Ubah code method main-nya. Program akan menanyakan dahulu apakah user akan main atau tidak. Jika iya maka permainan dimulai, jika tidak aplikasi akan ditutup.

```
public void run() {
FormUtama hangMan = new FormUtama();
hangMan.setVisible(true);
int start =
      JOptionPane.showOptionDialog(hangMan,
      "Mulai permainan?", "Mulai",
      JOptionPane.YES_NO_OPTION,
      JOptionPane.QUESTION_MESSAGE, null,
      null, null);
if (start == JOptionPane.YES_OPTION) {
hangMan.play();
} else {
System.exit(0);
}
}
```

Gambar 2.11

Code method main().

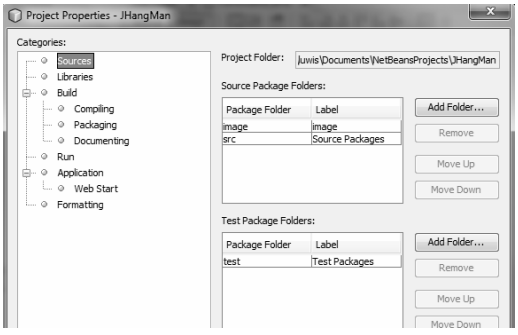


MENJALANKAN APLIKASI

1. Kita akan mengeset FormUtama sebagai main kelas project. Klik kanan pada project **JHangMan > Properties >** pilih Categories: **Run > Browse Main class >** pilih **FormUtama**.

Gambar 2.12

Mengeset FormUtama menjadi Main Class.



2. Jalankan project JHangMan. Klik menu **Run > Run main project** atau tekan F6.
3. Muncul tampilan seperti ini.

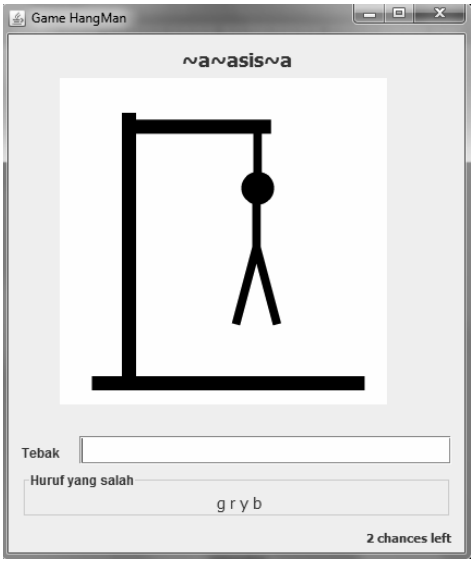
Gambar 2.13

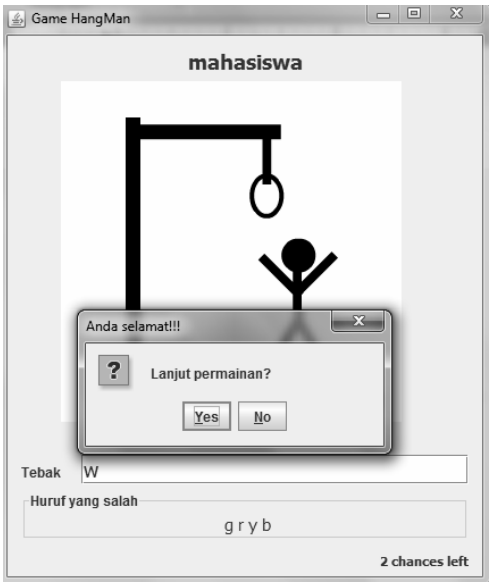
Memulai permainan.

4. Pilih **Yes** dan permainan akan dimulai. Ketikkan sebuah huruf untuk menebak kata yang dimaksud dan tekan **Enter**. Setiap tebakan huruf yang salah akan dicatat dan mengurangi jumlah kesempatan.

Gambar 2.14

Tampilan aplikasi HangMan.



Gambar 2.15

Penebakan kata berhasil.